Apresentação

O que é o shell

O shell é o "prompt" da linha de comando do Unix e Linux, é o servo que recebe os comandos digitados pelo usuário e os executa.

O shell é aquele que aparece logo após digitar-se a senha do usuário e entrar na tela preta. Ou na interface gráfica, ao clicar no ícone do Xterm, rxvt, Terminal ou Console.

```
localhost login: root
Password:

Last login: Fri Apr 16 01:57:28 on tty5
[root@localhost root]# _
```

Ali está o shell, esperando ansiosamente por algum comando para ele poder executar. Essa é a sua função: esperar e executar. Cada comando digitado é lido, verificado, interpretado e enviado ao sistema operacional para ser de fato executado.

No Mac OS X, o shell está em Aplicativos > Utilitários > Terminal. No Windows é preciso instalá-lo com o Cygwin.

Funcionando como uma ponte, o shell é a ligação entre o usuário e o kernel. O kernel é quem acessa os equipamentos (hardware) da máquina, como disco rígido, placa de vídeo e modem. Por exemplo, para o usuário ler um arquivo qualquer, toda esta hierarquia é seguida:

USUÁRIO --> SHELL --> KERNEL --> DISCO RÍGIDO

Para os usuários do Windows, é fácil pensar no shell como um MSDOS melhorado. Ao invés do C:\> aparece um [root@localhost root]#, mas o funcionamento é similar. Basta digitar um comando, suas opções e apertar a ENTER que ele será executado. O comando deve estar no PATH, mensagens de aviso são mandadas para a tela e Ctrl+C interrompe o funcionamento. Isso tudo é igual em ambos.

Mas o shell é muito mais poderoso que seu primo distante. Além dos comandos básicos para navegar entre diretórios e manipular arquivos, ele também possui todas as estruturas de uma linguagem de programação, como IF, FOR, WHILE, variáveis e funções. Com isso, também é possível usar o shell para fazer scripts e automatizar tarefas.

Este será o nosso foco: scripts em shell.

Shell script

Um script é um arquivo que guarda vários comandos e pode ser executado sempre que preciso. Os comandos de um script são exatamente os mesmos que se digita no prompt, é tudo shell.

Por exemplo, se de tempos em tempos você quer saber informações do sistema como horário, ocupação do disco e os usuários que estão logados, é preciso digitar três comandos:

```
[root@localhost root]# date
[root@localhost root]# df
[root@localhost root]# w
```

É melhor fazer um script chamado "sistema" e colocar estes comandos nele. O conteúdo do arquivo "sistema" seria o seguinte:

```
#!/bin/bash
date
df
w
```

E para chamar este script, basta agora executar apenas um comando:

```
[root@localhost root]# sistema
```

Isso é um shell script. Um arquivo de texto que contém comandos do sistema e pode ser executado pelo usuário.

Antes de começar

Se você está acessando o sistema como usuário administrador (root), saia e entre como um usuário normal. É **muito perigoso** estudar shell usando o superusuário, você pode danificar o sistema com um comando errado.

Se você não tem certeza qual o seu usuário, use o comando "whoami" para saber

Como o prompt de usuário normal é diferente para cada um, nos exemplos seguintes será usado "**prompt$**" para indicar o prompt da linha de comando.

O primeiro shell script

O primeiro shell script a fazer será o "sistema" do exemplo anterior, de simplesmente juntar três comandos em um mesmo script.

Passos para criar um shell script

1. Escolher um nome para o script

Já temos um nome: sistema.

 Use apenas letras minúsculas e evite acentos, símbolos e espaço em branco

2. Escolher o diretório onde colocar o script

Para que o script possa ser executado de qualquer parte do sistema, mova-o para um diretório que esteja no seu PATH. Para ver quais são estes diretórios, use o comando:

```
echo $PATH
```

 Se não tiver permissão de mover para um diretório do PATH, deixe-o dentro de seu diretório pessoal ($HOME).

3. Criar o arquivo e colocar nele os comandos

Use o nano, VI ou outro editor de textos de sua preferência para colocar todos os comandos dentro do arquivo.

4. Colocar a chamada do shell na primeira linha

A primeira linha do script deve ser:

```
#!/bin/bash
```

Para que ao ser executado, o sistema saiba que é o shell quem irá interpretar estes comandos.

5. Tornar o script um arquivo executável

Use o seguinte comando para que seu script seja reconhecido pelo sistema como um comando executável:

```
chmod +x sistema
```

Problemas na execução do script

"Comando não encontrado"

O shell não encontrou o seu script.

Verifique se o comando que você está chamando tem exatamente o mesmo nome do seu script. Lembre-se que no Unix/Linux as letras maiúsculas e minúsculas são diferentes, então o comando "SISTEMA" é diferente do comando "sistema".

Caso o nome esteja correto, verifique se ele está no PATH do sistema. O comando "echo $PATH" mostra quais são os diretórios conhecidos, mova seu script para dentro de um deles, ou chame-o passando o caminho completo.

Se o script estiver no diretório corrente, chame-o com um "./" na frente, assim:

```
prompt$ ./sistema
```

Caso contrário, especifique o caminho completo desde o diretório raiz:

```
prompt$ /tmp/scripts/sistema
```

"Permissão Negada"

O shell encontrou seu script, mas ele não é executável.

Use o comando "chmod +x seu-script" para torná-lo um arquivo executável.

"Erro de Sintaxe"

O shell encontrou e executou seu script, porém ele tem erros.

Um script só é executado quando sua sintaxe está 100% correta. Verifique os seus comandos, geralmente o erro é algum IF ou aspas que foram abertos e não foram fechados. A própria mensagem informa o número da linha onde o erro foi encontrado.